IMPLEMENTATION

```
/*Name:BHAGYA A JAI
Roll no:18
Experiment Name:CREATION OF THREADS*/
class Mythread extends Thread
{       public void run()
      {
        for(int i=1;i<=100;i++)
          {
            if(i%2!=0)
             System.out.println(i);
          }
      }
}
class Main
{       public static void main(String args[]) throws InterruptedException
     {  Mythread t=new Mythread();
          t.start();
          t.join();
          for(int j=1;j<=100;j++)
            {
             if(j%2==0)
             System.out.println(j);
            }
        }
}
```

OUTPUT

```
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
```

51
53
55
57
59
61
63
65
67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97
99
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54

56
58
60
62
64
66
68
70
72
74
76
78
80
82
84
86
88
90
92
94
96
98
100

IMPLEMENTATION

```
/*Name:BHAGYA A JAI
Roll no:18
Experiment Name:THREAD PRIORITIES*/
class Mythread extends Thread
{       static Thread mt;
      public void run()
       {
         for(int i=1;i<=10;i++)
           {   if(i%2!=0)
               System.out.println(i);
           }
       }
}
class Main
{       public static void main(String args[]) throws InterruptedException
      {  Mythread t=new Mythread();
           t.start();
           t.setPriority(1);
           Mythread.mt=Thread.currentThread();
           Mythread.mt.setPriority(2);
           System.out.println("Priority of Mythread:"+t.getPriority()+"\nAnd priority of main
thread:"+Mythread.mt.getPriority());
           for(int j=1;j<=10;j++)
             {
              if(j%2==0)
              System.out.println(j);
             }
        }
}
```

OUTPUT

```
Priority of Mythread:1
And priority of main thread:2
2
4
6
8
10
1
3
5
7
9
```

IMPLEMENTATION

```
/*Name:BHAGYA A JAI
Roll no:18
Experiment Name:MULTITHREADING APPLICATION*/
import java.util.Random;
class A implements Runnable
{       static int random;
      Random r=new Random();
      public void run()
      {  random=r.nextInt(25);
           System.out.println(random);
      }
}
class B implements Runnable
{       public void run()
      {
       if(A.random%2==0)
        System.out.println((int)Math.pow(A.random,2)+"\n");
      }
}
class C implements Runnable
{       public void run()
      {
       if(A.random%2!=0)
        System.out.println((int)Math.pow(A.random,3)+"\n");
      }
}
class Main
{       public static void main(String args[]) throws InterruptedException
      {  for(int i=0;i<10;i++)
           {     Thread a=new Thread(new A());
                 Thread b=new Thread(new B());
                 Thread c=new Thread(new C());
                 a.sleep(1000);
                 a.start();
                 b.start();
                 c.start();
           }
       }
}
```

OUTPUT

```
13
2197

4
16

14
196

24
576
```

0
0

19
6859

14
196

2
4

4
16

18
324

IMPLEMENTATION

```
/*Name:BHAGYA A JAI
Roll no:18
Experiment Name:THREADS USING RUNNABLE INTERFACE*/
class Mythread implements Runnable
{         public void run()
        {
          for(int i=1;i<=4;i++)
          System.out.println("In mythread");

       try
        {
          Thread.sleep(500);
        }
       catch(InterruptedException e)
        {
          System.out.println("Exception caught"+e);
        }
        }
}
class Main
{         public static void main(String args[]) throws InterruptedException
       {  Mythread mt=new Mythread();
             Thread t=new Thread(mt); //mt.join();
             t.start();
             t.join();
                   for(int i=1;i<=4;i++)
             System.out.println("Main thread");
          }
}
```

OUTPUT

```
In mythread
In mythread
In mythread
In mythread
Main thread
Main thread
Main thread
Main thread
```

IMPLEMENTATION

```
/*Name:BHAGYA A JAI
Roll no:18
Experiment Name:BANKING OPERATION USING MULTIPLE THREADS AND
SYNCHRONIZATION*/
class Demo1
{
        static double bal;
        public synchronized void bank()
        {
        System.out.println("To DEPOSIT:1000");
                bal=bal+1000;
                System.out.println("Current balance after deposit of 1000:"+bal);

                System.out.println("To WITHDRAW:500");
                Demo1.bal=Demo1.bal-500;
                System.out.println("Current balance after withdrawal of 500:"+bal);

        }
}
class Demo2 extends Thread
{     Demo1 d;
      Demo2(Demo1 d)
          { this.d=d;
          }
      public void run()
          {       d.bank();
          }
}
class Main
{
        public static void main(String args[])
        {
                Demo1 de1=new Demo1();
                Demo2 d1=new Demo2(de1);
                Demo2 d2=new Demo2(de1);
                d1.start();
                d2.start();
        }
}
```

OUTPUT

With synchronisation

```
To DEPOSIT:1000
Current balance after deposit of 1000:1000.0
To WITHDRAW:500
Current balance after withdrawal of 500:500.0
To DEPOSIT:1000
Current balance after deposit of 1000:1500.0
To WITHDRAW:500
Current balance after withdrawal of 500:1000.0
```

Without synchronization
To DEPOSIT:1000
To DEPOSIT:1000
Current balance after deposit of 1000:2000.0
To WITHDRAW:500
Current balance after deposit of 1000:2000.0
To WITHDRAW:500
Current balance after withdrawal of 500:1500.0
Current balance after withdrawal of 500:1000.0